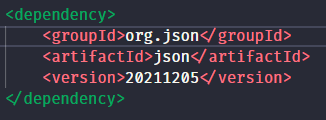
**Respostas Teste Técnico Junior**

Primeiramente, clone o repositório <https://gitlab.com/glauco-pacto/padawan-teste-v2>  Clonado o projeto, você terá uma aplicação em java, onde usamos o framework Spring Boot, e as dependências são gerenciadas com o maven.

**RESPOSTA:** <https://github.com/HumbertoGaldino/padawan-teste-v2>

1 - **Yoda**: Como primeira parte de seu treinamento, capaz você deverá ser para subir a aplicação;

**RESPOSTA:** Arquivo pom.xml atualizado com dependência do pacote org.json  
  


2 - Qual será o retorno se eu fizer um **POST** em <http://localhost:9099/generic-post> passando o seguinte body?

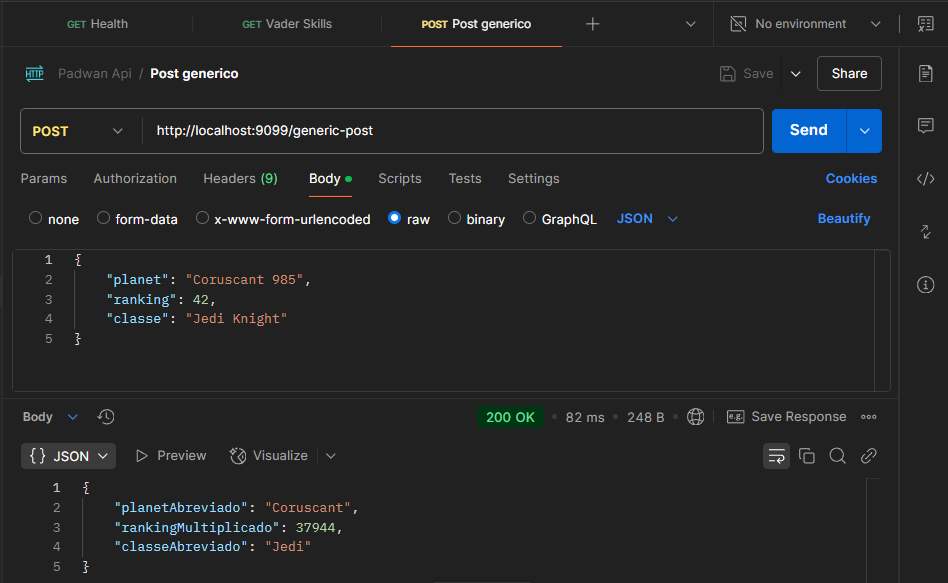
**RESPOSTA:**

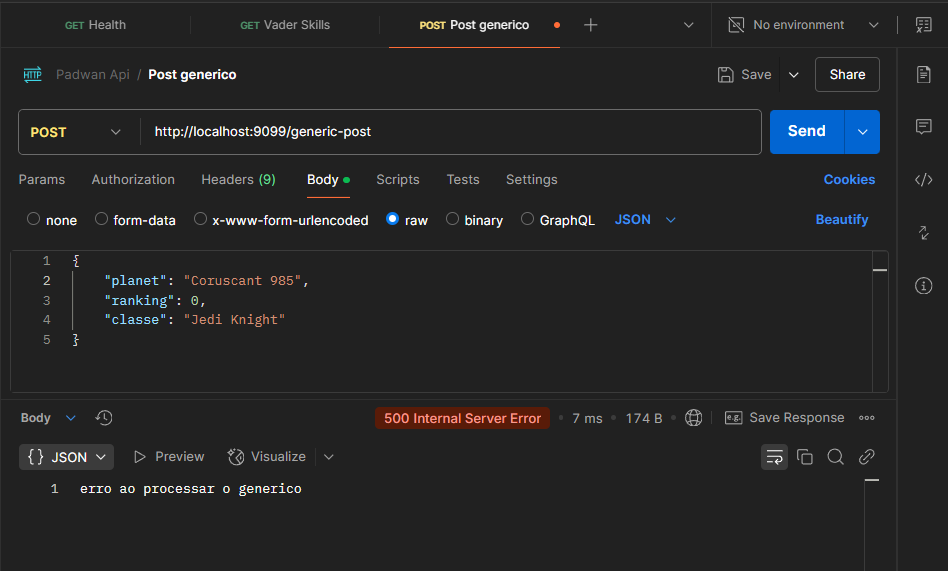
{

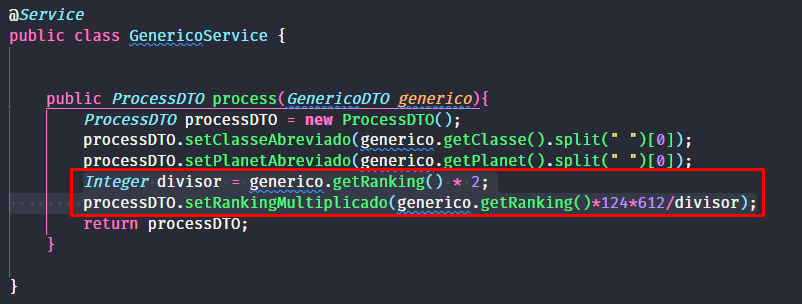
"planetAbreviado": "Coruscant",

"rankingMultiplicado": 37944,

"classeAbreviado": "Jedi"

}  
  


3 - Se mudarmos o campo “ranking” para 0, a aplicação dá um erro. Por quê?  
RESPOSTA:  
  




A resposta “erro ao processar o genérico” ocorre devido ao valor de entrada ser 0, fazendo com que a variável divisor, mesmo tendo multiplicado o valor do ranking por 2, receba o valor de 0. Sendo assim ao executar a próxima linha o processo é interrompido porque não existe divisão por 0.

4 - Como podemos mudar a porta onde a aplicação sobe?  
**RESPOSTA:** Alterando a variável server.port no arquivo application.properties inserido na pasta resources que se encontra dentro de main.

5 - Agora você vai ter que programar. Massa, hein?! Quando acessamos o endpoint <http://localhost:9099/vader/skills>, a aplicação retorna uma lista de habilidades do Darth Vader. Faça com que o sistema tenha um endpoint igual a esse, só que com seu nome e suas habilidades. Por exemplo, o meu seria [http://localhost:9099/joaoalcides/skills](http://localhost:9099/vader/skills). Detalhe muito importante: **ao fazer isso, você não pode impactar o do Darth Vader.**

**RESPOSTA:** Foi criado um novo service – HumbertoService.java – para retorno das skills. Logo após criei um controller – HumbertoController.java – para criação do endpoint e seu retorno.   
  
**6 - Mestre Windu** explica que todo Jedi já foi um Padawan, então necessário é definir nome, status (padawan/jedi/mestre jedi) e um mentor para cada Jedi;

**7 - Qui-Gon** ao fazer contato, estando no mundo espiritual da Força, nos diz que é possível mensurar a quantidade de midichlorians em cada Jedi, por isso, você deverá informar isso lá;

**8 - Anakin Skywalker** antes de se tornar o Darth Vader, deixou para você a última missão, após isso, estará apto a se tornar um Jedi. Essa missão é a de criar, com Native SQL as seguintes consultas, com seus respectivos endpoints:

* 1. Listar todos os mestres Jedis e seus aprendizes;
  2. Listar todos Jedis cujo midichlorians sejam acima de 9000;
  3. Listar por categoria, quantos são os Jedis;

**6, 7 e 8 RESPOSTA:**Foi criado uma model – Jedi.java – para coleta dos dados. Logo após criei o controller – JediController.java – para criação dos endpoints de coleta e inserção dos dados. Criei também um repository – JediRepository.java – para utilização do Native SQL.